
CLASS ATTENDANCE SYSTEM USING **FACIAL RECOGNITION**

Group - 4

Nafisa Ayman Prome 2012027642

Zia Uddin Chowdhury 2013754042

Abstract

A person's identity is represented by the distinctiveness or uniqueness of their face. In this project an individual's face is used to automatically track attendance. The student's attendance is important for any college, university, and institution. Calling a student by name or roll number and recording their attendance is the traditional technique of taking attendance. Time consumption for this procedure is in an important point of concern. For example if a class is held for one hour to record the students it requires 5 to 10 minutes or if the attendance recording system is held by RFID card swapping then if a student forget to bring the card to the class he/she might miss the attendance. To prevent these losses an automatic process is used in this project which is based on image processing. In this project face detection and facial recognition is used. Face detection is used to locate the position of face region and face recognition is used for compare the image of a student with previous dataset - the recorded images of the students in this case

Table of Contents

Title and info	1
Abstract	2
Table of Contents	2
Introduction	3
Background and Related Work	3
Problem Identification	4
Solution Methodology	4
Screenshots of the progress	6
Expected Timeline	8
Gantt Chart and the progress	10
Methodologies and the technologies required	10
Future Scope	11
Conclusion	11
References of Articles and YouTube videos	11

Introduction

In the era of 21st century, we have witnessed various adoption of digital technologies, and monitoring class attendance via digital systems is one of them. Face recognition liveness detection attendance systems may totally automate the manual or biometric monitoring of the attendance of the students.

Background and Related Work

Facial recognition is a technique that measures an individual's physical or behavioural characteristics for identification purposes. In today's world, facial recognition is very widely used which is an advanced, automated, and sensible identification system that can identify a person by facial features. It uses a digital camera to capture the image along with a computer for processing and analysis, and finally an excel sheet to output the attendance data of the person.

It is straightforward to identify any person from different facial images and it is a high-speed and reliable technology. Attendance system using facial recognition has a wide array of uses in offices, firms, banks and educational institutions for recording the presence of the employees and students. It can also be used in automated teller machines, retail banking, airport check-in and customer identification.

Our system would comprise of a mobile application responsible for clicking pictures which would interact with the model. The AI model would then process the image, recognising the student and logging information such as - Name, ID and time of the attendance into an excel sheet. These sheets would then be uploaded automatically to Google Drive via a python script when the time of the attendance would be over, ideally within the first 20 minutes of the class.

Problem Identification

There are various downsides to the traditional way of manual attendance as described below:

- ❖ Hassle of taking the manual attendance
 - For a class of vast number of students, the process of manually ticking off attendance can be a daunting task and can be deemed as inefficient and time consuming. Using our facial recognition system, the students would be able to quickly give their own attendance.
- ❖ Exact time of the attendance wouldn't be known
 - In case of the manual attendance, the exact time of the attendance wouldn't be known. So even if a student arrives at the class prior to the finishing, he/she would be marked as present.
- ❖ RFID system is not reliable as anyone could punch the card
 - Anyone can give the attendance using the RFID card but by using the facial recognition system, the attendance would only be given only if the student is present and scans himself.
- ❖ It is costly to develop attendance system hardware and software
 - The cost of developing an advanced attendance system would require huge investment due to the placement of the hardwares and the development of the softwares. But using our system, all it takes is a simple mobile application for marking the attendance.
- ❖ Chances of missing the attendance for large number of people in class
 - For large groups of people in the class, it can be a common scenario of missing out the attendance when it is called. Our automated system would tick off the attendance just by the recognition of the face during the entry of the student in the class.
- ❖ Hassle of carrying ID/punch card
 - Often it is seen that people forget to bring the ID cards, and miss out on the attendance. Using our system, the students would be able to let go of the hassle of carrying the ID card.

Solution Methodology

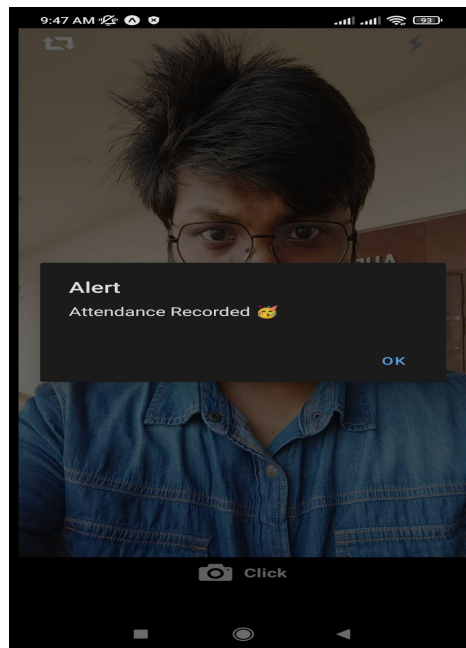
- ❖ Getting rid of manual attendance
 - For a group of a large number of people, it is quite a daunting task to tick off the manual attendance and there might be a source of error. While entering the classroom, the students would be able to scan their faces using our mobile application, which would send the data over to the machine learning model. The model uses OpenCV for facial recognition and can easily identify the student.
- ❖ Quick processing of the image
 - The process needs to be faster to avoid long waiting queues of the students, so we would develop the application just for the purpose of clicking an image. The image would then be sent over to the model for processing.
- ❖ Exact time of the attendance
 - Our system would login the exact details of the student as well the time when the attendance is given. The system would maintain a time of 20 minutes, from the beginning of the class to record the student data, so the late comers wouldn't be excused as the file would be uploaded automatically to Google Drive.
- ❖ Comparatively a reliable system
 - Students might miss the attendance call or the RFID card could be punched by friends, making it unreliable. Our system would detect the faces of the students to provide the attendance, hence making it mandatory for the student to be present.
- ❖ Cost effective system
 - We would only require to set up a local server for the processing of data and develop a mobile application, which can be used to click pictures. This minimizes the hardware cost involved in other technologies such as the RFID system.
- ❖ No hassle of carrying an ID/punch card
 - The system would just require the presence of the student and there would be no need for carrying an ID/punch card for the attendance

The initial setup of how the program compared two images

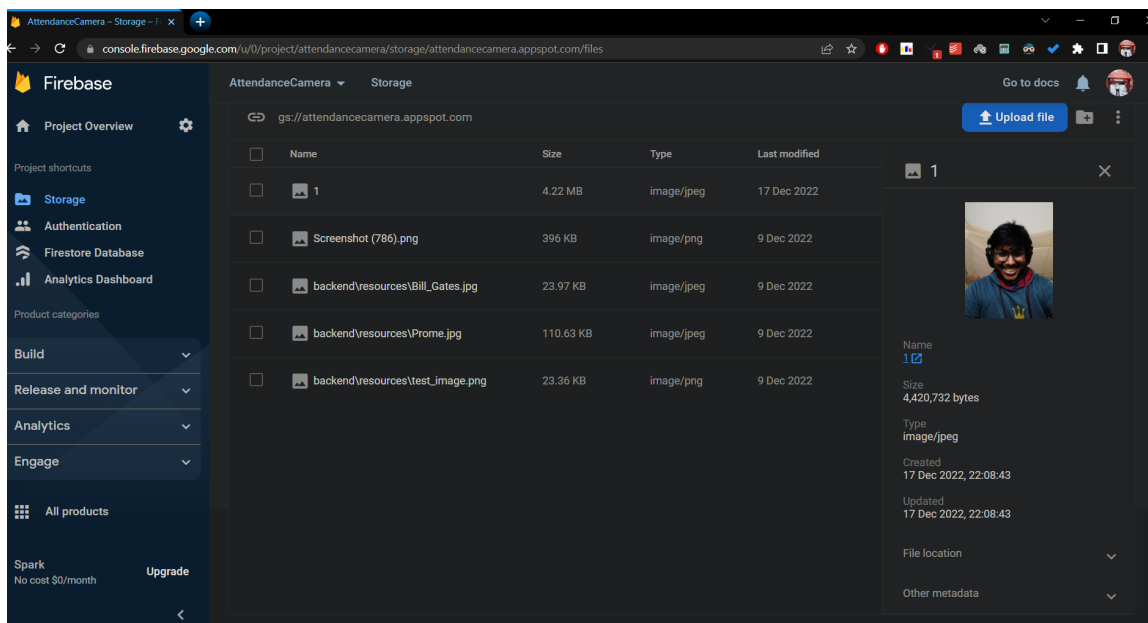
React Native's Expo Camera framework was used to develop the camera app. This is how the UI of the app looks like, which would be used to click the picture.



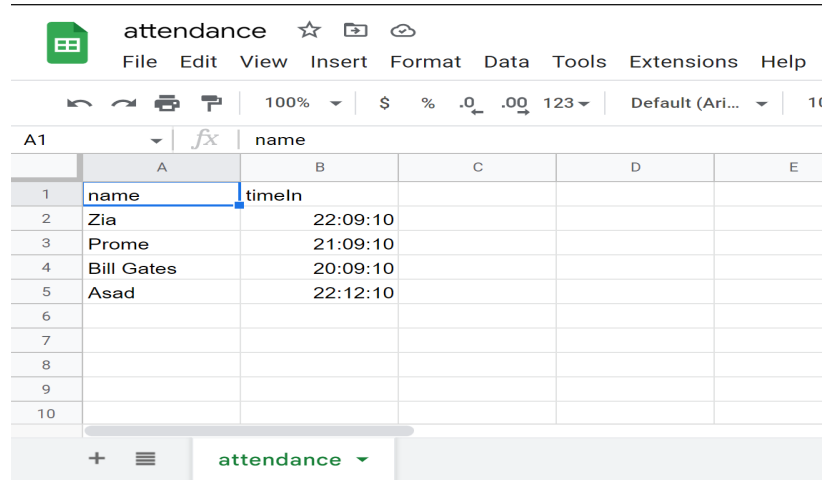
- ❖ After the image is being clicked, there would be options to send the image to the server for the attendance or to retake the image. The attendance would be given once the server finishes the comparison.



- ❖ The image would be temporarily stored in firebase storage and then it would be sent over to the PC servers for marking the attendance.



- ❖ Python scripts on the server would be able to automate the task of retrieving the image from firebase, compare it with the server dataset and mark the attendance. The updated attendance sheet would then be uploaded to Google Drive.



The screenshot shows a Google Sheets interface with a spreadsheet titled 'attendance'. The spreadsheet has two columns: 'name' and 'timeIn'. The data is as follows:

	A	B	C	D	E
1	name	timeIn			
2	Zia	22:09:10			
3	Prome	21:09:10			
4	Bill Gates	20:09:10			
5	Asad	22:12:10			
6					
7					
8					
9					
10					

Expected Timeline

Week 1: A brief detailing of the class attendance system using facial recognition and the technologies to be used.

Week 2: Setting up the VS code environment with Python and the required libraries along with CMake for the compilation. Setting up OpenCV library for the facial recognition.

Week 3: Gather the sample images and comparing those using OpenCV facial recognition library. The program correctly compared the images and displayed whether the images matched or not and estimated the distance between the two images. The face of the sample images were automatically detected using a rectangular box.

Week 4: Adding the webcam feature to the program and it was able to detect the face of a person. The face was then automatically compared with the sample images. If a match is found, the system is able to identify the person and display their name. The name and the time of the attendance would be logged in to the excel sheet, if not already done for the person. If the person tries again, the time won't still be updated as his/her attendance has been previously recorded.

Week 5: Setting up the android development environment in VS code along with the react-native libraries. Setting up an android emulator for functioning accordingly as an alternate to an android device. Using react-native framework - Expo or react-native-vision-camera for the android camera app. The app would run successfully

on the emulator and android device. The button for clicking the image will be added to the app.

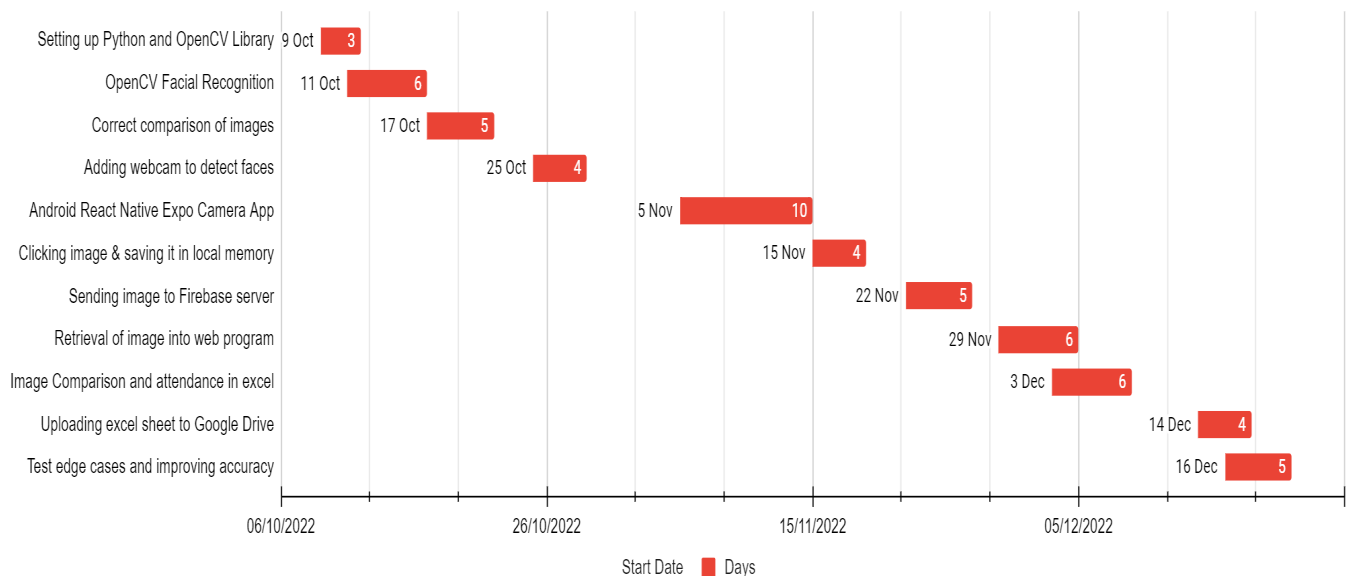
Week 6: The android app would be able to click images and save it successfully in a memory location along with the feature of sharing the image.

Week 7: Google Authentication and API implementations for accessing the Google Drive. Development of a python script for automatically uploading the excel sheet to Google Drive, 20 minutes after the class has begun.

Week 8: The android app will be able to send the images to the backend server using an API and the image would be compared to the sample images. The result would then be processed as before and stored in the Excel sheet.

Week 9: Handling edge cases and feeding more sample images to the model, so that accuracy is increased. Rigorous testing of the combination of the backend server and the mobile app, so that the face detection is as accurate as possible.

Gantt Chart



Methodologies and the technologies required

The technologies that were used are as follows:

- 1) **Python 3.10.8** - used as the backbone of the backend server. All of the required libraries are imported and installed. The coding environment was setup in Visual Studio Code.
<https://www.python.org/>
- 2) **OpenCV** - for the processing of the images and the results derived after the comparison of the clicked image from the android application is sent to the backend server.
<https://opencv.org/>
- 3) **Android** - The android app was developed using React Native and the camera application was made using Expo Camera framework which had the features of capturing the image using the front and the back camera as well as uploading the image to Firebase storage.
<https://reactnative.dev/>
<https://docs.expo.dev/versions/latest/sdk/camera/>
- 4) **Firebase** - The image clicked by the android device was temporarily stored in Firebase storage, to be downloaded using python into the backend server, to be sent for comparison.
<https://firebase.google.com/>
- 5) **Google Drive API and authentication** - Permissions and authentication of the Google Drive API from Google Playground and using it for automatically uploading the excel sheet in Drive.
<https://developers.google.com/oauthplayground/>

Our system comprised of a python backend server and a mobile application. The student would be able to click his/her image using the camera app made using React Native and Expo Camera. The app would then be used to send the image to the firebase storage, storing it temporarily. The backend server in python would consist of the programs responsible for fetching the image from the firebase storage and storing it on the local server. The image would then be processed using OpenCV and comparisons made with the dataset of the students of the class. It also consists of the program which then update the attendance of the student in the excel sheet along with the time of the attendance and automatically uploads it to Google Drive. The attendance would be stopped after 20 minutes of the commencement of the class. All of the programs would work together in sync, automatically updating and uploading the data, so that the system works efficiently.

Future Scope

- ❖ It can be extended further to take attendance not only in schools and universities but also in banks, offices and other workplaces.
- ❖ Depth(Third dimension) can be introduced to the image processing phase, so that the system is able to differentiate between a 2D (camera) image and 3D (actual person).
- ❖ A model can be setup on Google collab, based on the dataset collected to provide much accurate results based on the learning and comparison.

Conclusion

Student Attendance System using Facial Recognition is a powerful tool, which would be a bold step in the field of technological advancement. Such a system would be beneficial for both the students and the faculty likewise as it would save a lot of time and hassle of manual attendance.

References of Articles and YouTube videos

https://www.youtube.com/watch?v=sz25xxF_AVE&t=3s&ab_channel=Murtaza%27sWorkshop-RoboticsandAI
<https://brainhub.eu/library/opencv-react-native-image-processing>
<https://www.geeksforgeeks.org/connect-your-android-phone-camera-to-opencv-python/>
https://www.youtube.com/watch?v=4WPjWK0MYMI&ab_channel=MissCoding
https://www.youtube.com/watch?v=1dcFNPA1sTw&t=2s&ab_channel=ProgrammingwithMash
<https://www.timedynamo.com/blog/face-recognition-attendance-system>